Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.176 – 182

RESEARCH ARTICLE

Analysis of Malware Detection Techniques in Android

Ms. Prajakta D. Sawle[1]
M.E First Year CSE
P.R. Patil COET, Amravati,
Maharashtra, INDIA
pdsawle@gmail.com

Prof. A. B. Gadicha[2]
Department of Information Technology
P .R. Patil COET, Amravati
Maharashtra, INDIA
ajjugadicha@gmail.com

Abstract-The malware threat for mobile phones is expected to increase with the functionality enhancement of mobile phones. This threat is increased with the surge in population of smart phones instilled with stable Internet access which provides attractive targets for malware developers. Currently, in the smartphone market, Android is currently the most popular smartphone operating system. Due to this popularity and also to its open source nature, Android-based smartphones are now an ideal target for attackers. Since the number of malware designed for Android devices is increasing fast, Android users are looking for security solutions aimed at preventing malicious actions from damaging their smartphones. Anti-malware products promises to effectively protect against malware on mobile devices and many products are available for free or at reasonable prices. From this perspective, we propose and analyse some potential limitation-oriented techniques for effective malware detection.

Keywords: - Smartphone, Malware detection, Android malware, smartphone security, Anti-malware

I. INTRODUCTION

Smartphone usage has been rapidly increasing and it is increasingly becoming a sophisticated device. This increasing popularity makes the attackers more attracted to these devices. Smartphone use is now not just limited to personal conversation but has expanded to financial transactions, internet banking and for storing personal data. This has made smartphones more vulnerable to malware attacks and a target for information and identity theft.

Researchers from Kaspersky Lab first found the malware called Cabire, for mobile phone in 2004[1]. This paper discusses about analysis of different mobile malware detection techniques.

II. MALWARE

The malware are termed as malicious software that is designed specifically to target a mobile device system, such as a tablet or smartphone to damage or disrupt the device. Most mobile malware is designed to disable a mobile device, allow a malicious user to remotely control the device or to steal personal information stored on the device [3]. Once malware gets itself into the system by different media like copying of files from external devices onto the system and mostly by downloading files from the internet, it checks the vulnerabilities of the

system and infects the system if the system is highly vulnerable. The concern for the rate of spread of malware today is a global phenomenon, especially as it spreading double over the internet which is a means of global communication. Today's malware is capable of doing many things, such as: stealing and transmitting the contact list and other data, locking the device completely, giving remote access to criminals, sending SMS and MMS messages etc. Mobile malware causes serious public concern as the population of mobile phones is much larger than the population of PCs [2].

III. MOTIVES TO CREATE MOBILE MALWARE

A. Novelty and Amusement

Some malware causes mischief or damage in a way that appears to be intended to amuse the author. For example, Ikee. Changed the wallpaper of infected iPhone devices, and sent anti-religion text messages text messages from Android phones. Many pieces of malware fall into this category and no other [4].

B. Selling User Information

Mobile operating system APIs provide applications with large amounts of information about users. Applications can query mobile APIs for the user's location, list of contacts, browser and download history, list of installed applications, and IMEI (the unique device identifier). Although we cannot know for sure why malware collects this information, we hypothesize that this data is being sold by malware distributors for financial gain. Advertising or marketing companies might be willing to purchase users' locations, browsing histories, and lists of installed applications to improve behavioural profiling and product targeting. However, advertising libraries in legitimate applications already routinely collect user location, and web-based advertisements already track browsing habits.

C. Stealing User Credentials

Credentials could be used directly by malware authors for greater financial gain, but financial fraud can be difficult to perpetrate and requires specialization. People use smartphones for shopping, banking, e-mail, and other activities that require passwords and payment information. Banks rely on cell phones for two-factor authentication. Users may also save authentication and payment credentials in text documents on their phones (for example, to use the phone as a mobile password manager). This makes cell phones a target for credential theft. Three pieces of malware in our data set target user credentials by intercepting SMS messages to capture bank account credentials [4].

D. Premium-Rate Calls and SMS

Legitimate premium-rate phone calls and SMS messages deliver valuable content, such as stock quotes, technical support, or adult services. The cost of a premium-rate call or SMS is charged to the sender's phone bill. Premium rate calls can cost several dollars per minute, and premium-rate SMS messages can cost several dollars per message. In Android and Symbian, malware can completely hide premium-rate SMS messages from the user. Premium-rate SMS attacks could feasibly go unnoticed until the user's next phone bill[4].

E. SMS Spam

SMS spam is used for commercial advertising and spreading phishing links. Commercial spammers are incentives to use malware to send SMS spam because sending SMS spam is illegal in most countries. Sending spam from a compromised machine reduces the risk to the spammer because it obscures the provenance of the spam. Furthermore, the use of SMS may lend more authenticity to spam than e-mail because phone contacts are often more intimately acquainted than e-mail contacts. 8 of the malicious Symbian and Android applications send SMS spam [4].

F. Search Engine Optimization

Many web sites rely on search engines for traffic, which makes web site owners desire high visibility in search engine results. Search engines rank web sites according to how relevant each web site is to a given search term. An engine's perception of relevance is influenced by the rate at which users click on the web sites returned for a search term. A web site will rise in the results for a search term if many people search for that term and then click on that web site. Malware can be employed to improve a website's ranking in search engine results. This type of malware sends web requests to the search engine for the target search term. The malware then

fraudulently clicks" on the search result that corresponds to the target web site. As a result, the web site's rank for that search term will increase [4].

G. Ransom

Malware can be a tool for blackmail. For example, the desktop Trojan Kenzero stole the user's browser history, published it publicly on the Internet alongside the person's name, and then demanded 1500 yen to take down the person's browser history. There has not yet been any mobile malware that seriously threatens or publicly embarrasses the user for profit, but one piece of mobile malware has sought a ransom. A Dutch worm locked iPhone screens and demanded 5 euros to unlock the screens of infected phones [4].

IV. MALWARE ATTACK TECHNIQUES

The infection strategies of malware include entry point obfuscation, code integration, code insertion, register renaming, memory access reordering and session hijacking. In entry point obfuscation, the virus hijacks the control of the program after the program has been launched, overwrite program import table addresses and function call instructions. During the code integration, a virus merges its code with legitimate program that requires disassembly of target which is a very difficult operation (W95/Zmist). Malware can also either append virus code and thereby modify the entry point of a legitimate program or inject its code into unused sections of a program code. On the other hand, malware has two basic strategies adopted on a cell phone viz;

1) By creating a new process to launch its attack

2) By redirecting the program flow of a legitimate application in order to execute its malicious code within a legitimate security context (e.g messaging process)[5].

A. Malware First Attack Technique on Mobile Phone

Malware, in this case created a new process to execute its malicious code and compromise the cell phone. This is a case where user operations are required, for example when a user downloads software on an internet or opens a received message from another user. The newly created process contains a program descriptor, which describes the address content, execution state and security context, which is different from that of the invoked parent process. This technique is widely adopted by the most existing malware one to its simplicity. In this technique, the cell phone malware launch an attack through legally installed application, having realized that the Symbian and windows programs register themselves within a platform and use their system services within their API framework. A good example is a cardblock Trojan, which is a cracked version of a legitimate Symbian application called instansis. It allows a user to create SIS archive. Cardblock blocks the MMC memory card and detect the subdirectories under system(SDI attack)[6].

B. Malware Second Attack Technique on Mobile Phone

Malware, in this case redirects the program flow of a legitimate application (e.g. messaging activities) to execute its malicious code within a legitimate security context [5]. Open Source based OS and application a framework is the major target of this kind of malware attack. i.e Android smart phones. This type of attack is possible for malware by exploiting the stack buffer overflow in a Linux-based cell phone to “hijack “the normal program flow and launch its attacks [6].

V. MALWARE METHOD OF PROPAGATION

The basic method of propagation of malware is either self-propagation or user interaction. A malware like worm does not require any user intervention before its execution occur. It is capable of copying itself and causing occasional execution without the intervention of host program or its user. Virus on the other hand is a user-interaction oriented malware that always looks for a host program for its execution and consequent infection. Other malware might not require any of these methods for its propagation, but may adopt internet medium for their spreading. Mobile malware on the other hand, adopt mobile phone network on the internet in order to propagate itself, but this action is usually curtailed by the internally built defense mechanism in the network mobile phone. Another opportunity for mobile malware to propagate is through the direct pair-wise communication resources i.e. Bluetooth, Wi-Fi, and Infrared [7].

VI. MALWARE DETECTION TECHNIQUES

The task of detecting malware can be categorized into analysis, classification, detection and eventual containment of malware. Several classification techniques have been used in order to classify malware according to their instances and this has made it possible to recognize the type and activities of a malware and new variant. Analysis of malware has to do with identifying the instances of malware by different classification

schemes using the attributes of known malware characteristics. Malware detection has to do with the quick detection and validation of any instance of malware in order to prevent further damage to the system. The last part of the job is containment of the malware, which involves effort at stopping escalation and preventing further damages to the system. A commercial antivirus uses signature based technique where the database must be regularly updated in order to possess the latest virus data detection mechanisms. However, the zero-day malicious exploit malware cannot be detected by antivirus, based on signature-based scanner, but the use of statistical binary content analysis of file to detect anomalous file segments [9]. Toward this end, malware detection technique has been categorized according to the following:

A. Signature-Based malware detection

Signature-based detection works by scanning the contents of computer files and cross-referencing their contents with the "code signatures" belonging to known viruses. A library of known code signatures is updated and refreshed constantly by the anti-virus software vendor. If a viral signature is detected, the software acts to protect the user's system from damage. Suspected files are typically quarantined and/or encrypted in order to render them inoperable and useless. Clearly there will always be new and emerging viruses with their own unique code signatures. So once again, the anti-virus software vendor works constantly to assess and assimilate new signature-based detection data as it becomes available, often in real time so that updates can be pushed out to users immediately and zero-day vulnerabilities can be avoided.

A pattern-marching approach commercial antivirus is an example of signature based malware detection where the scanner scans for a sequence of byte within a program code to identify and report a malicious code. This approach to malware detection adopts a syntactic level of code instructions in order to detect malware by analysing the code during program compilation. This technique usually covers complete program code and within a short period of time. However, this method has limitation by ignoring the semantics of instructions, which allows malware obfuscation during the program's run-time.

B. Specification-based malware detection

Specification based detection makes use of certain rule set of what is considered as normal in order to decide the maliciousness of the program violating the predefined rule set. Thus programs violating the rule set are considered as malicious program. In specification-based malware detection, where a detection algorithm that addresses the deficiency of pattern-matching was developed. This algorithm incorporates instruction semantics to detect malware instances. The approach is highly resilience to common obfuscation techniques. It used template T to describe the malicious behaviours of a malware, which are sequence of instructions represented by variables and symbolic constants. The limitation of this approach is that the attribute of a program cannot be accurately specified. Specification-based detection is the derivate of anomaly based detection. Instead of approximating the implementation of a system or application, specification based detection approximates the requirements of application or system. In specification-based system there exists a training phase which attempts to learn the all valid behaviour of a program or system which needs to inspect. The main limitation of specification based system is that it if very difficult to accurately specify the behaviour the system or program. One such tool is Panorama which captures the system wide information flow of the program under inspection over a system, and checks the behaviour against a valid set of rule to detect malicious activity [8] .

C. Behavioural-based Detection

The behaviour-based malware detection system is composed of several applications, which together provide the resources and mechanisms needed to detect malware on the Android platform. Each program has its own specific functionality and purpose in the system and the combination of all of them creates the Behavior-Based malware detection system. The Android data mining scripts and applications mentioned in are the responsible for collecting data from Android applications, and the script running on the server will be the responsible for parsing and storing all collected data. Furthermore, the script will be responsible for creating the system call vectors for the k-means clustering algorithm.

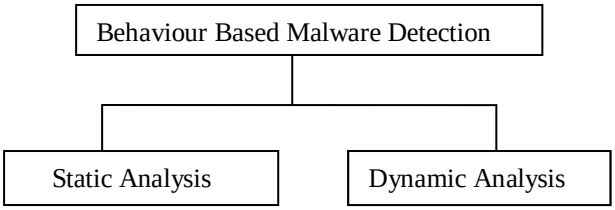

Figure1: Static and Dynamic Analysis

The methods of analysis of the behavior-based malware detection system developed in this project can be divided into two main groups: Static Analysis and Dynamic Analysis.

Static Analysis is responsible for analyzing Android source code _les in order to _nd malicious code patterns or signatures. This form of analysis will decompress, disassemble and search for patterns in the APK files. The method is fast and does not generate a high processing load.

Dynamic Analysis analyzes the behavior of Android applications by monitoring system calls with the Strace tool. All input traces generated by the Android smartphone user will be collected using the data collector application well as the crowd sourcing and data collector script. In Dynamic Analysis the user will install, execute and generate input data for the Android applications in order to obtain an application behavior output log file.

Table1: Static and Dynamic Malware Analysis Advantages and disadvantages

	Advantages	Disadvantages
Static Analysis	Cheap and Fast. Not very resource consuming	Have to know Malware patterns Or signatures in patterns
Dynamic Analysis	Detection of unknown attacks	Highly resource consuming, not Feasible for Battery Devices

D. APPLICATION PERMISSION ANALYSIS

Applications run in a sandbox environment however they need permissions to access certain data. At the time of installation, Android platform asks the user to grant or deny permission for the application based on the activities the application can perform. Section 4.4.2 has more description about the permission based security in Android devices. In 2009 Enck et al. [14] proposed Kirin security service for Android platform, to authorize an application to perform sensitive activities. This is to overcome a limitation in Android platform where the developers can intentionally hide permission label to a component. If no label is specified there is no restriction as it had *default allow* policy. The Kirin security service interacts with Android Application installer and it also interacts with collection Kirin Security rules. Rules represent the malicious patterns and it is compared with configuration of the installed application. The study proposes five steps to identify dangerous configurations – (1) Check the phone‟s assets, (2) What are the functional requirements, (3) Analyse asset security goals and threats (4) Specify security requirements (5) Analyse security mechanism limitations[10].

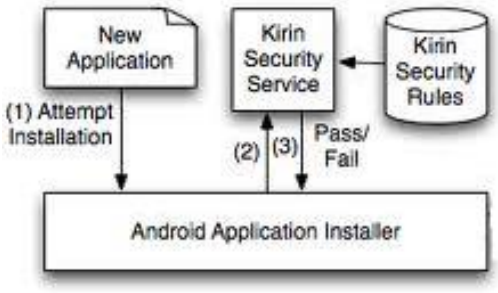

Figure2: Kirin Security System

E. Cloud Based Malware Detection

Google Play applications are scanned for malware. Google uses a service named Bouncer to automatically scan applications on the Google Play Store for malware. As soon as an application is uploaded, Bouncer checks it and compares it to other known malware, Trojans, and spyware. Every application is run in a simulated environment to see if it will behave maliciously on an actual device. The applications behaviour is compared to the behaviour of previous malicious apps to look for red flags. New developer accounts are particularly scrutinized – this is to prevent repeat offenders from creating new accounts Google Play can remotely uninstall applications: If you've installed an app that is later found to be malicious, Google has the ability to remotely uninstall this application from your phone when it's pulled from GooglePlay. Google announced an exciting security feature called the "application verification service" to protect against harmful Android applications. As stated in a recent Google+ post by a member of the Google Android team, "Now, with Jelly Bean Android 4.2 devices that have Google Play installed have the option of using Google as an application verifier. We will check for potentially harmful applications no matter where you are installing them from". Google to directly face Android malware threats and take such measures to better protect Android users. When you install the Android 4.2 update on your system, you'll be greeted with a pop-up notification asking you if you want to verify all installed apps. If you ignore that popup, you can also enable appVerification from the Settings > Security menu.

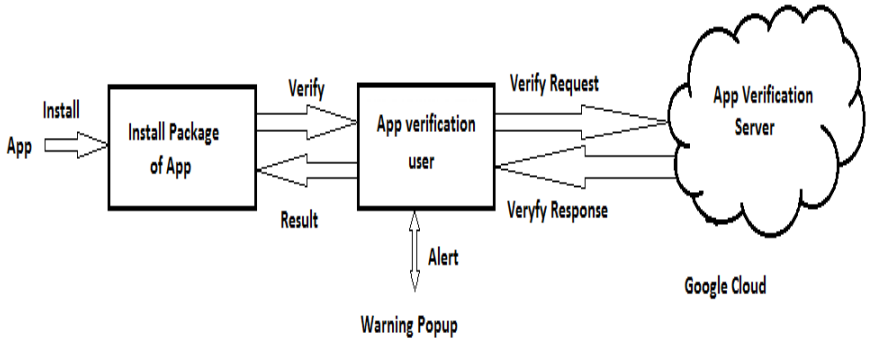

Figure3: Cloud-Based Detection System

A user can turn the service on/off by going to "Settings," "Security," and then "Verify apps." When an app isbeing installed (Step 1), the service, if turned on, will be invoked (Step 2) to collect and send information about the app (e.g., the app name, size, SHA1 value, version, and the URL associated with it) as well as information about the device (e.g., the device ID and IP address) back to the Google cloud (Step 3). After that, the Google cloud will respond with a detection result (Step 4). If the app is not safe, the user is then shown a warning popup (Step 5) flagging the app as either dangerous or potentially dangerous. Dangerous apps are blocked from being installed, while potentially dangerous ones instead alert users and provide an option to either continue or abort the installation (Step 6) with a warning popup.

Figure4: Enabling the app verification service

Figure5: Detecting a potentially dangerous app

Figure6: Detecting a dangerous app

Android 4.2 scans side loaded apps: While apps on Google Play are checked for malware, apps that are side loaded (installed from elsewhere) were not checked for malware. On Android 4.2, when you first try to sideload an app, you'll be asked whether you want to verify side loaded apps are safe. This ensures that all apps on your device are checked for malware[8].

F. Social collaboration

In 2011, Yang et al. [13] have illustrated a new malware detection architecture based on social collaboration and used the concept of *hot set*. The study focused on improving existing cloud-based solutions. In the cloud based system discussed in the previous Section, additional hardware for centralized servers and device emulators are required. The hot set concept states that not all malware signatures are equally important. To improve the performance, the hot-set is kept in the phone memory. Each mobile will store the *hot set* signatures for local detection and depend on other social group of mobile device users for *cold sets*.

This approach is termed as Social-AV. The idea is to have a portion of the full signature database in a device and rely on their social group to have a complete signature database. The hot-set in the device is kept up to date with latest signatures and to effectively manage it, it can adopt *Least Frequently Used* and *Least Recently Used* replacement techniques. Moreover, the size of the hot-set is made configurable to enable randomness in hot-sets in the entire devices in social group. The study found out that collaboration based approach enhanced the efficiency by 55% when compared with existing Antivirus systems.

VII. CONCLUSION AND FUTURE DIRECTIONS

The sharp increase in the number of smartphones on the market, with the Android platform posed to becoming a market leader makes the need for malware analysis on this platform an urgent issue. All market indicators forecast a massive increase in the number of smartphones purchased in the next 5 years. This will create a potential for a massive increase in malware generation, and in particular in the sector dominated by the market leader, potentially the Android platform. We illustrate various detection techniques proposed by various researchers. The presented detection techniques are viable, but large scale testing is required to determine real world performance. As Android malware evolves the effectiveness of these measures will decrease.

The next step is to deploy the Crowdroid lightweight client on Google's Android market and distribute it to as many users as possible. Users running our application will be able to see their own smartphone behaviour. We could even alert the users when one of their applications shows an abnormal trace. The system can also act as an early warning system, capable of detecting malicious or abnormally behaving applications in the early stages of propagation.

REFERENCES

[1] Aswathy Dinesh Ming dinesh@tufts.edu ,ChowAswathy.”An analysis of mobile malware and detection techniques”.
[2] Marwa M. A. Elfattah, Aliaa A.A Youssif, Ebada Sarhan Ahmed,” Handsets Malware Threats and Facing Techniques”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 12, 2011.
[3]http://www.webopedia.com/TERM/M/mobile_malware.html.
[4] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steven Hanna, and David Wagner, “A Survey of Mobile Malware in the Wild”1stACM workshop on Security and privacy in smartphones and mobile devices, October 2011.
[5] Abhijit, B., Xin, H., Kang G. S. and Taejoon, P. (2008) “ Behavioral detection of Malware on Mobile Handsets”, June 17–20, 2008,Breckenridge, Colorado, USA. ACM 978-1- 60558-139-2/08/06.
[6] Adebayo, Olawale Surajudeen, Mabayoje, Amit Mishra, Osho Oluwafemi, “Malware Detection, Supportive Software Agents and Its Classification Schemes”, International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.6, November 2012.
[7] Gjergji, Z., Goefrey M., Michael, L., and Per, J. (2005) “Defending Mobile Phones fro Proximity Malware”
[8] Vinit B. Mohata “Mobile Malware Detection Techniques” International Journal of Computer Science & Engineering Technology (IJCSET) ,2013
[9] Markus Jakobsson, Karl-Anders Johansson,” Retroactive Detection of Malware With Applications to Mobile Platforms”, HotSec'10 Proceedings of the 5th, USENIX conference on Hot topics in security, Article No. 1-13, USENIX Association Berkeley, CA, USA ©2010.
[10] Srikanth Ramu” Mobile Malware Evolution, Detection and Defense ” EECE 571B, TERM SURVEY PAPER, APRIL 2012.
[11] Liu, L. G., Zhang, Y, X., Chen. S. “VirusMeter: Preventing your cellphone from spies” In Proceedings of RAID, volume 5758 of Lecture Notes in Computer Science, 2009.
[12] Hahnsang Kim , Joshua Smith , Kang G. Shin, Detecting energy-greedy anomalies and mobile malware variants, Proceeding of the 6th international conference on Mobile systems, applications, and services, June 17-20, 2008, Breckenridge, CO, USA
[13] Yang, Liu; Ganapathy, Vinod; Iftode, Liviu; "Enhancing Mobile Malware Detection with Social Collaboration" Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference, 2011
[14] William Enck , Machigar Ongtang , Patrick McDaniel, "On lightweight mobile phone application certification", Proceedings of the 16th ACM conference on Computer and communications security, November 09-13, 2009, Chicago, Illinois, USA

